CS3511 Lab Report

Name: Adam O'Brien
Student ID: 117385301

Q1:
6-7 characters.
10 digits.
26 lowercase-letters.
26 upper-letters.

6: 10*52*62*62*62*62 =7,683,694,720
7: 10*52*62*62*62*62*62 =476,389,072,640
8: 10*52*62*62*62*62*62*62 =29,536,122,503,680

=30,020,195,271,040 combinations

Q2:

Prepared statements help defend against SQL injection attacks because the parameters, that are passed later don't have to be escaped. If the original statement does not come from external input an attack can't occur.

Not using prepared statements:

```
<?php
$firstname = $_GET["fname"];
$lastname = $_GET["lname"];
$email = $_GET["email"];
// Any value could be passed as a variable even Drop table MyGuests;
// Data would have to be escaped.
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ($firstname, $lastname, '$email)";

$conn->query($sql);
$conn->close();
?>
```

Using prepared statements:

```
<?php]
// prepare and bind
```

```
// Insert ? where we want to substitute a string, integer, Boolean or blob
$stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname, email) VALUES (?, ?, ?)");

// The sss argument tells SQL what the datatypes are.
//By telling SQL the data to expect we minimise the risk of injection attacks.
$stmt->bind_param("sss", $firstname, $lastname, $email);

// set parameters and execute
$firstname = $_GET[“fname”];
$lastname = $_GET[“lname”];
$email = $_GET[“email”];
$stmt->execute();
$stmt->close();
$conn->close();
?>
```

Q3

Electronic Code Book(ECB) is the simplest of encryption modes. A drawback of ECB is the lack of diffusion due to the fact it encrypts identical plain text blocks into identical cipher text blocks.

Formula:	**Ciphertext**
$Y_i=F(PlainText,Key)$	Y_i

Because of this it does not hide data patterns well and does not ensure message confidentiality. This leaves the data susceptible to replay attacks since every block is decrypted the exact same way.

Q4

Stored Cross Site Scripting:
The injected script is permanently stored on the targets server such as a database. The malicious script is then retrieved when the user requests the stored information.

Reflective Cross Site Scripting:
The injected script is reflected on the web server in a response that includes some or all of the input sent to the server as part of the request. When a user is tricked into clicking a malicious link, submitting a form or browsing a malicious site, the injected code travels to the vulnerable web site, reflecting the attack back on the user’s browsers which executes the code.

Q5

Cross Site Request Forgeries are prevented by the inclusion of an unpredictable token in each HTTP request. Each token at a minimum should be unique to each user session. Include the token in a hidden field so it is sent in the HTTP as opposed to the URL where it is more prone to exposure.
Requiring users to reauthenticate via a CAPTCHA can also be used to prevent CSRF.

Token Pattern:
A CSFR token should be:
Unique per user session.
A large random value.
Generated by a Crypto Graphically Secure Pseudo-Random Number Generator.

e.g.
<form action="/
transfer.do
" method="post
">
<
input type="hidden" name="
CSRFToken
"
value="OWY4NmQwODE4ODRjN2Q2NTlhMmZlYWEwYzU1YWQwMTVhM2JmNGYx
YjJiM
GI4MjJjZDE1ZDZMGYwMGEwOA
==">
...
</form>